Assignment 8: The Digital Fourier Transform

Jnaneswara Rao Rompilli [EE20B052]

April 16, 2022

1 Introduction

This assignment is about DFT and how it is implemented in python using Numpy's FFT module. We also attempt to approximate the continuous time fourier transform of a gaussian by windowing and sampling in time domain, and then taking the DFT.

2 Tasks

2.1 FFT and IFFT

We find the Fourier transform and invert it back to the time domain for a random signal, find maximum error to test the reconstruction

```
# FFT of random values
x = np.random.rand(100)
X = fft(x)
y = ifft(X)
np.c_[x, y]
print(abs(x - y).max())
```

2.2 Spectrum of sin(5t)

The phase for some values near the peaks is non zero. To fix this we sample the input signal at an appropriate frequency. We also shift the phase plot so that it goes from $-\pi$ to π.

```
# f(t) = sin(5*t)
x = np.linspace(0, 2 * PI, 129)
x = x[:-1]
y = np.sin(5 * x)
Y = fftshift(fft(y)) / 128.0
w = np.linspace(-64, 63, 128)

# Magnitude and phase plot of above function
plt.figure(1)
plt.subplot(2, 1, 1)
plt.plot(w, abs(Y), lw=2)
plt.xlim([-10, 10])
plt.ylabel(r"$|Y|$", fontsize=10)
plt.title(r"Spectrum of $\sin(5t)$", fontsize=12)
plt.grid(True)
plt.subplot(2, 1, 2)
plt.plot(w, np.angle(Y), "ro", lw=2)
ii = np.where(abs(Y) > 1e-3)
plt.plot(w[ii], np.angle(Y[ii]), "go", lw=2)
plt.xlim(-10, 10)
plt.ylabel(r"Phase of $Y$", fontsize=10)
plt.xlabel(r"$k$", fontsize=10)
plt.grid(True)
```

As expected we get 2 peaks at +5 and -5 with height 0.5

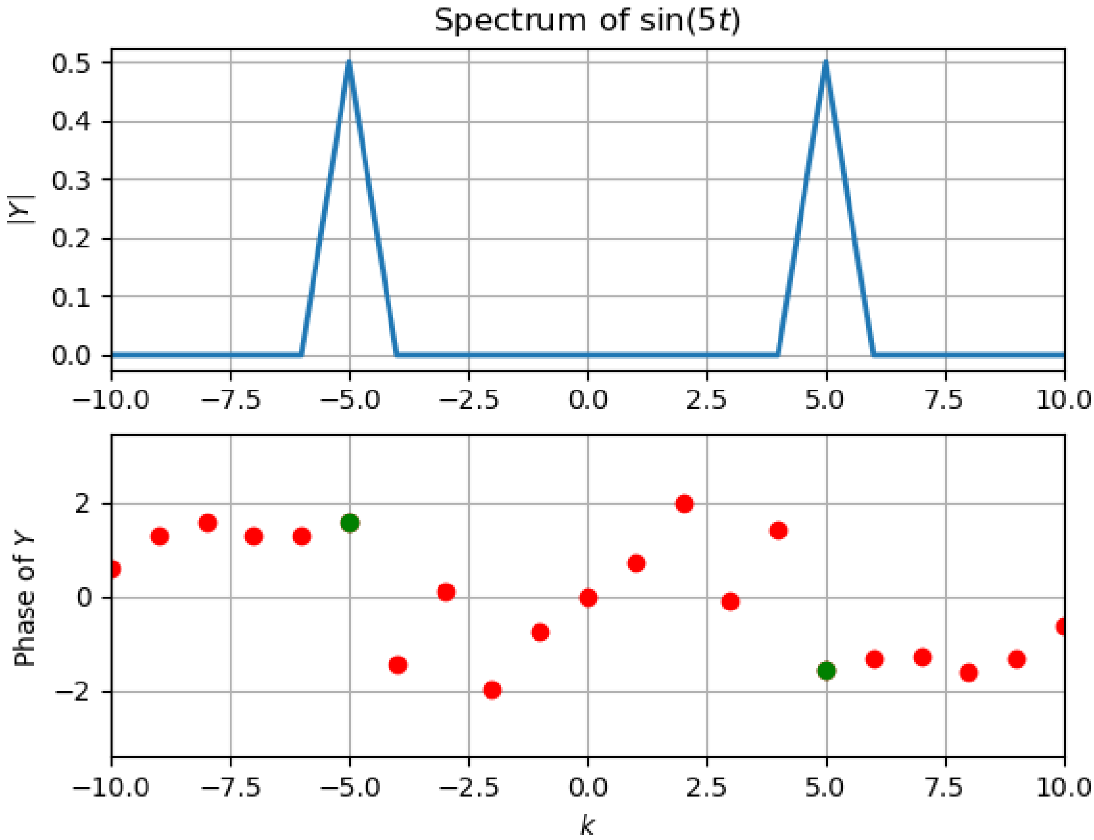

Figure 1: Magnitude and Phase Plot of sin(5t)

2.2.1 Amplitude Modulation

Consider the signal:

$$f(t) = (1 + 0.1\cos(t))\cos(10t) \tag{1}$$

We expect a shifted set of spikes, with a main impulse and two side impulses on each side. This is because,

$$0.1cos(10t)cos(t) = 0.05(cos11t + cos9t) = 0.025(e11tj + e9tj + e11tj + e9tj) \tag{2}$$

The Python code to calculate the DFT for above function is as follows:

```
# f(t) = (1 + 0.1cos(t))*cos(10t)
t = np.linspace(-4 * PI, 4 * PI, 513)
t = t[:-1]
y = (1 + 0.1 * np.cos(t)) * np.cos(10 * t)
Y = fftshift(fft(y)) / 512.0
w = np.linspace(-64, 63, 513)
w = w[:-1]

# Magnitude and phase plot of above function
plt.figure(2)
plt.subplot(2, 1, 1)
plt.plot(w, abs(Y), lw=2)
plt.xlim([-15, 15])
plt.ylabel(r"|Y|", fontsize=10)
plt.title(
    r"Spectrum of $\left(1+0.1\cos\left(t\right)\right)\cos\left(10t\right)$",
    fontsize=12,
)
```

```
plt.grid(True)
plt.subplot(2, 1, 2)
plt.plot(w, np.angle(Y), "ro", lw=2)
plt.xlim([-15, 15])
plt.ylabel(r"Phase of $Y$", fontsize=10)
plt.xlabel(r"$\omega$", fontsize=10)
plt.grid(True)
```

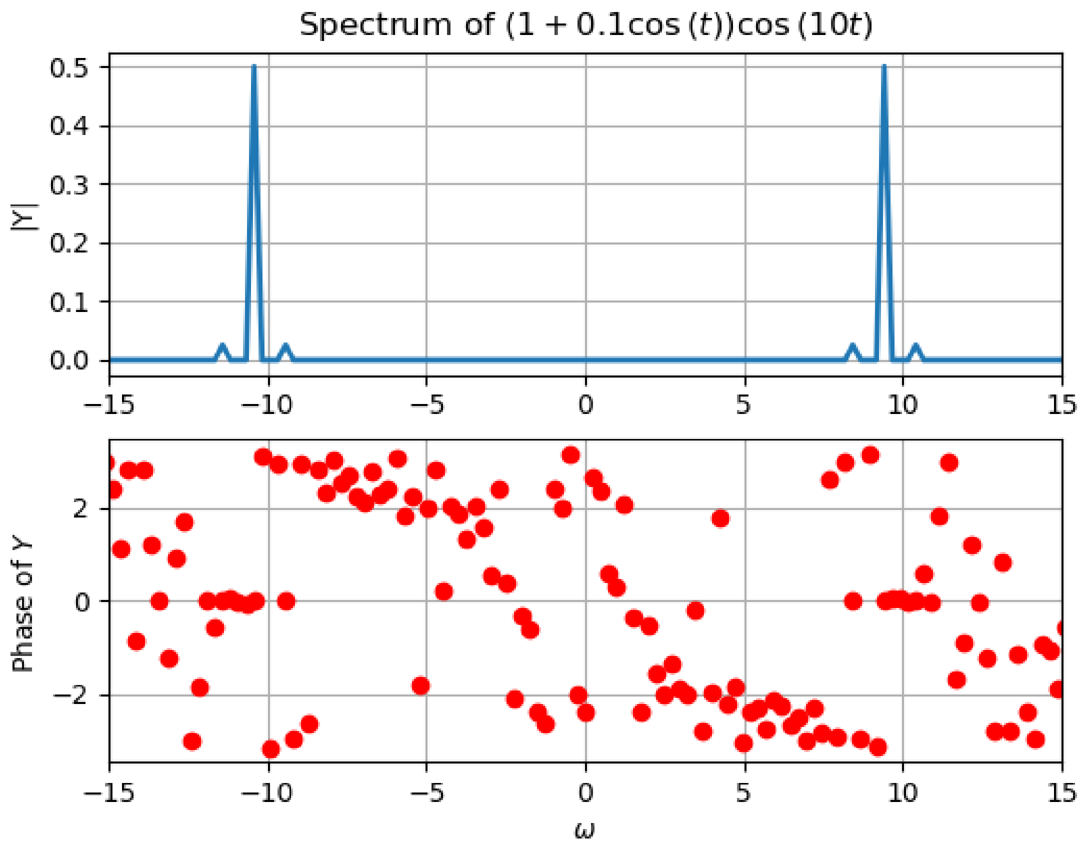

Figure 2: Magnitude and Phase Plot of (1 + 0.1cos(t))*cos(10t)

2.2.2 Spectrum of $sin^3(t)$

This signal can be expressed as a sum of sine waves using this identity:

$$\sin^3(t) = \frac{3}{4}\sin(t) - \frac{1}{4}\sin(3t) \tag{3}$$

The Python code for above function is as follows:

```
# f(t) = sin^3(t) = (3*sin(t) - sin(3*t)) / 4
t = np.linspace(-4 * PI, 4 * PI, 513)
t = t[:-1]
y = np.sin(t) * np.sin(t) * np.sin(t)
Y = fftshift(fft(y)) / 512.0
w = np.linspace(-64, 64, 513)
w = w[:-1]

# Magnitude plot of above function
plt.figure(3)
plt.plot(w, abs(Y), lw=2)
plt.xlim([-10, 10])
```

```
plt.xlabel(r"$\omega$", fontsize=10)
plt.ylabel(r"|Y|", fontsize=10)
plt.title("Spectrum of sin^3(t)", fontsize=12)
plt.grid(True)
```

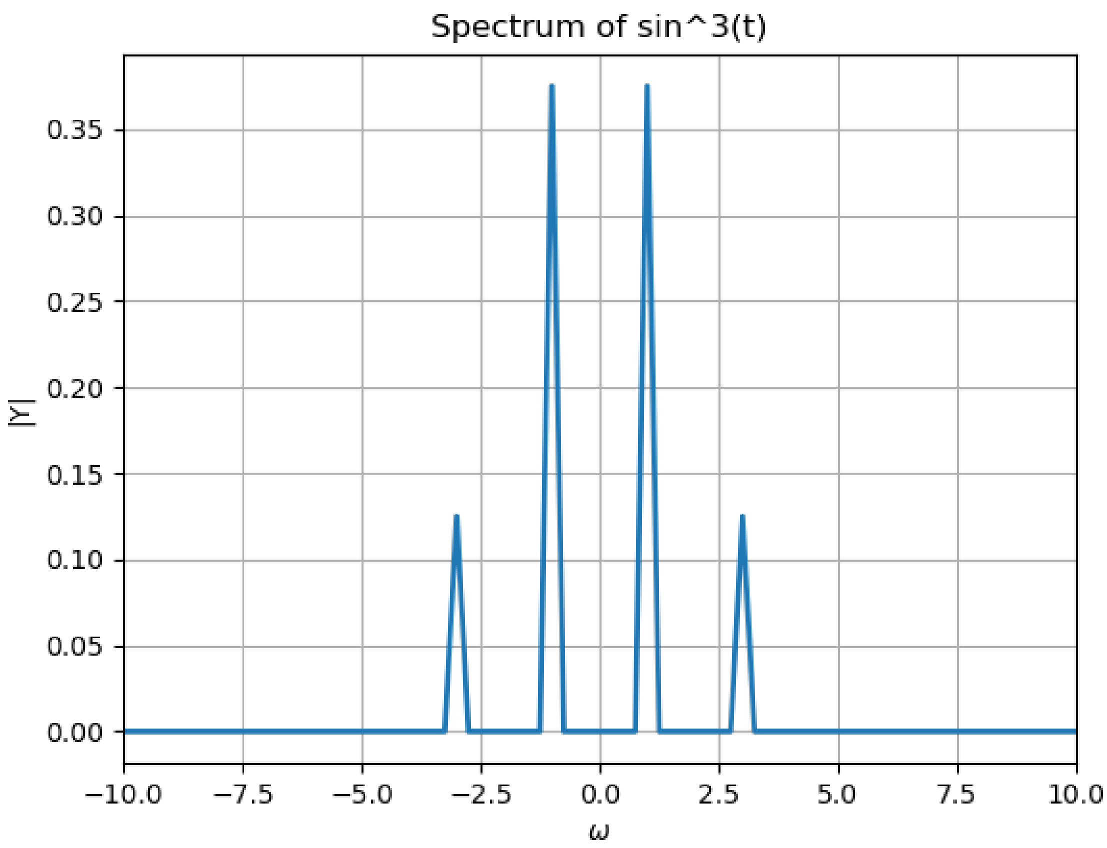

Figure 3: Magnitude Plot of $sin^3(t)$

2.2.3 Spectrum of $cos^3(t)$

This signal can be expressed as a sum of cosine waves using this identity:

$$sin^3(t) = \frac{3}{4}\cos(t) + \frac{1}{4}\cos(3t) \tag{4}$$

The Python code for above function is as follows:

```
# f(t) = cos^3(t) = (3*cos(t) + cos(3*t)) / 4
t = np.linspace(-4 * PI, 4 * PI, 513)
t = t[:-1]
y = np.cos(t) * np.cos(t) * np.cos(t)
Y = fftshift(fft(y)) / 512.0
w = np.linspace(-64, 64, 513)
w = w[:-1]

# Magnitude plot of above function
plt.figure(4)
plt.plot(w, abs(Y), lw=2)
plt.xlim([-10, 10])
plt.ylabel(r"|Y|", fontsize=10)
plt.xlabel(r"$\omega$", fontsize=10)
plt.title("Spectrum of cos^3(t)", fontsize=12)
plt.grid(True)
```

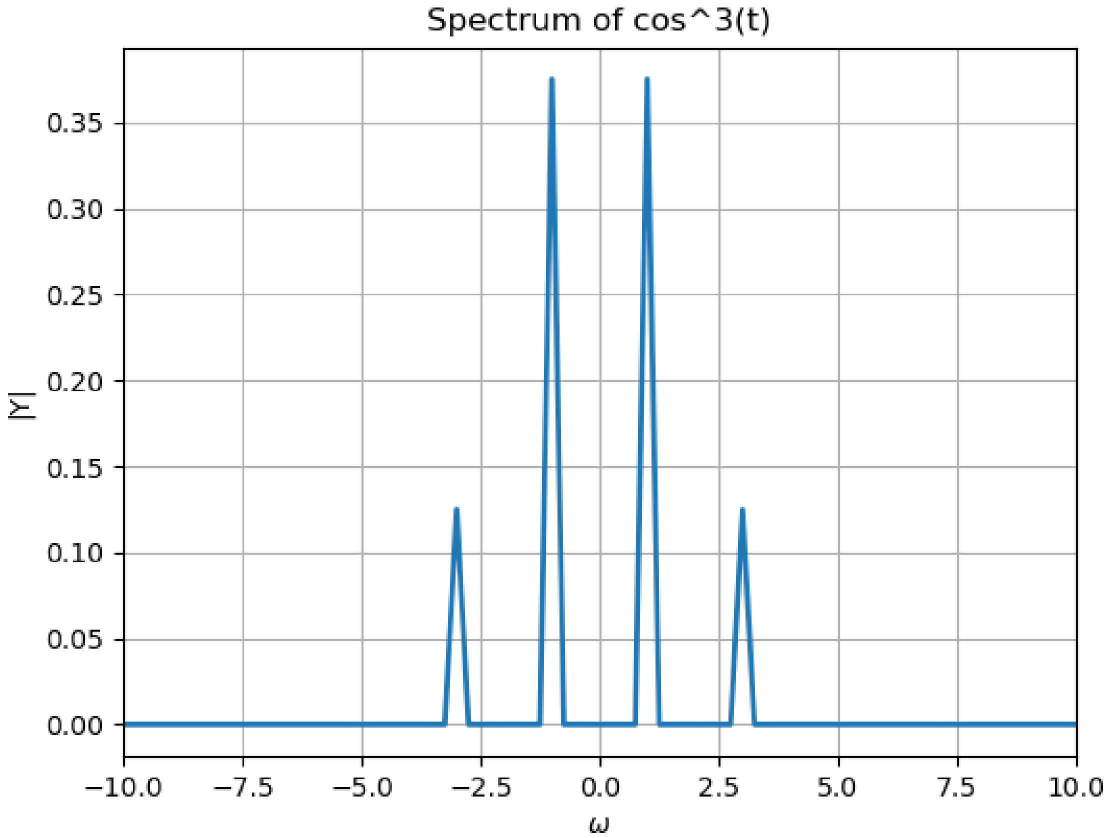

Figure 4: Magnitude Plot of $cos^3(t)$

2.3 Spectrum of frequency Modulated wave

Consider the signal:

$$f(t) = \cos(20t + 5\cos(t)) \tag{5}$$

The Python code for above function is as follows:

```
# f(t) = cos(20t + 5cos(t))
t = np.linspace(-4 * PI, 4 * PI, 513)
t = t[:-1]
y = np.cos(20 * t + 5 * np.cos(t))
Y = fftshift(fft(y)) / 512.0
w = np.linspace(-64, 64, 513)
w = w[:-1]

# Magnitude and phase plot of above function
plt.figure(5)
plt.subplot(2, 1, 1)
plt.plot(w, abs(Y), lw=2)
plt.xlim([-40, 40])
plt.ylabel(r"$|Y|$", fontsize=10)
plt.title(r"Spectrum of $\cos(20t + 5cos(t))$", fontsize=12)
plt.grid(True)
plt.subplot(2, 1, 2)
ii = np.where(abs(Y) > 1e-3)
plt.plot(w[ii], np.angle(Y[ii]), "go", lw=2)
plt.xlim(-40, 40)
plt.ylabel(r"Phase of $Y$", fontsize=10)
```

```
plt.xlabel(r"$\omega$", fontsize=10)
plt.grid(True)
```

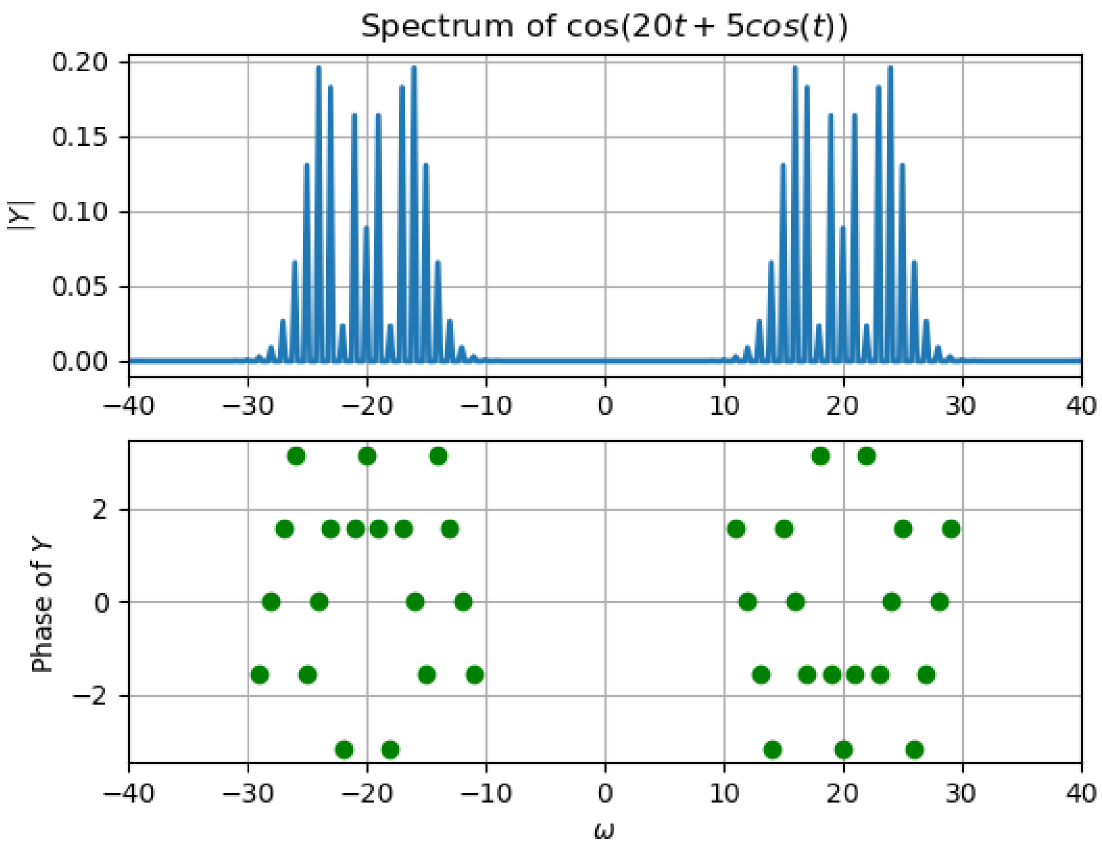

Figure 5: Magnitude Plot of $cos(20t + 5cos(t))$

2.4 Spectrum of Gaussian Function

The Fourier transform of a signal $x(t)$ is defined as follows:

$$X(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{6}$$

As Gaussian Function tends to 0 for large values of t, we can approximate the above equation as shown below. And the appropriate values of T will be calculated using iterations

$$X(\omega) \approx \frac{1}{2\pi}\int_{-T/2}^{T/2} x(t)e^{-j\omega t}dt \tag{7}$$

The expression for the Gaussian is :

$$x(t) = e^{\frac{-t^2}{2}} \tag{8}$$

The CTFT is given by:

$$X(j\omega) = \frac{1}{\sqrt{2\pi}}e^{\frac{-\omega^2}{2}} \tag{9}$$

`The Python code for above function is as follows:`

```
# f(t) = exp(-t*t/2)
N = 128
T = 2 * PI
tolerance = 1e-15
```

```
error = 1
w = []
Y = []
# Only when the error is less than a tolerance value will the loop be terminated
while True:
    t = np.linspace(-T / 2, T / 2, N + 1)
    t = t[:-1]
    w = N / T * np.linspace(-PI, PI, N + 1)
    w = w[:-1]

    y = np.exp(-0.5 * t * t)
    Y = fftshift(fft(y)) * T / (2 * PI * N)
    Y_act = (1 / np.sqrt(2 * PI)) * np.exp(-0.5 * w * w)

    error = np.mean(abs(abs(Y) - Y_act))

    if error < tolerance:
        break

    T, N = 2 * T, 2 * N

print(f"For accurate frequency domain: N = {N}, T = {T/PI}*")
# Magnitude plot of above function
plt.figure(6)
plt.plot(w, abs(Y), lw=2)
plt.xlim([-10, 10])
plt.ylabel(r"$|Y|$", fontsize=10)
plt.xlabel(r"$\omega$", fontsize=10)
plt.title("Spectrum of Gaussian function", fontsize=12)
plt.grid(True)
```

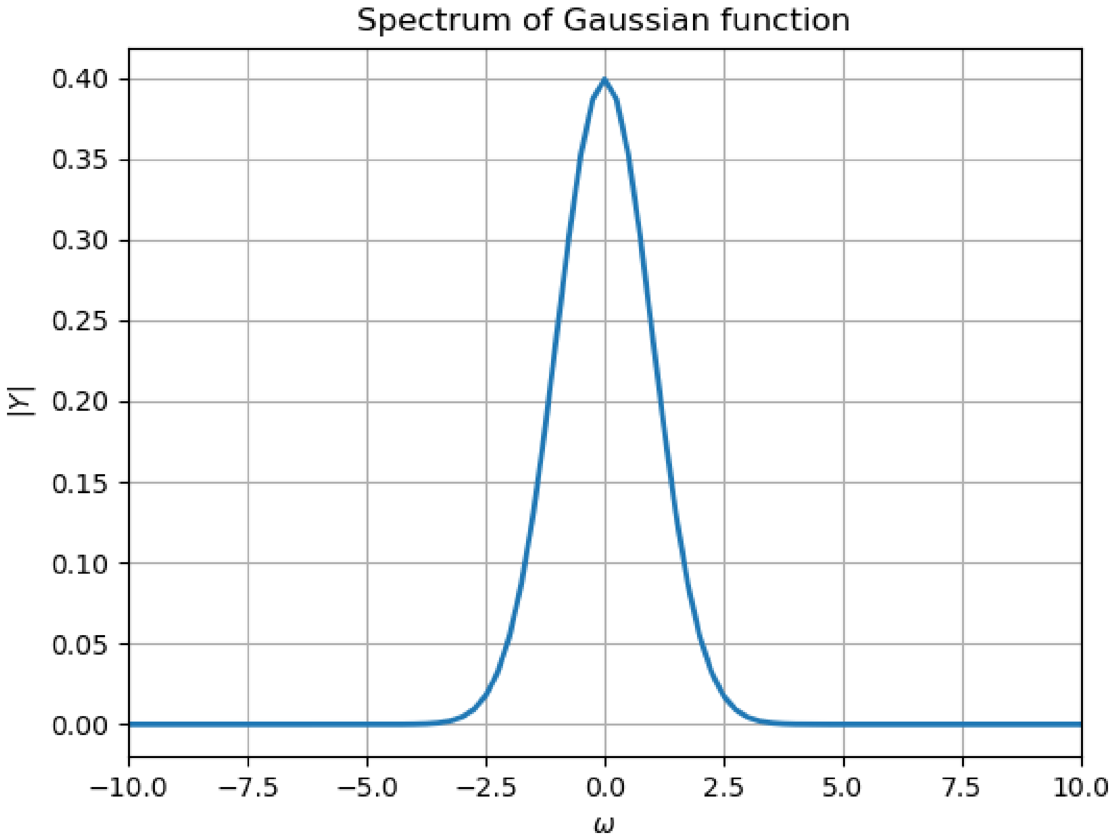

Figure 6: Magnitude Plot of $e^{\frac{-t^2}{2}}$

3 Conclusion

We analysed the Discrete Fourier Transforms of sinusoids, amplitude and frequency modulated signals using FFT library in python. The pure sinusoids contained impulses at given frequncies. The frequency modulated wave contains many frequencies. Also we verified that DFT of a gaussian is gaussian function in w